

SKIDMARKS



SOFTWARE



"The racingest,
chasingest multi-
player racing game
ever!"

ALL AMIGAS 1MB RAM
3 DISKS + FREE AGA DISK

Precautions

1. If you are playing for a long time change control methods to avoid repetitive strain injury. Also drive in pairs, allowing one driver to sleep while the other races.
2. The disks are incredibly sensitive. Do not place in microwave oven or shower with them. Unlike the Amiga Format cover disk the disks included in this package will not fit in your ear.
3. Surgeon General's Warning: Do not make copies of the track disks for your friends or attempt to crack the protection system. Having your finger nails removed with a large pair of pliers may affect your ability to play SkidMarks.
4. Not registering your copy of SkidMarks will mean you won't receive upgrades to your game including more trackdisks, more cars and fixes to the software.

What You Get

The SkidMarks package includes this manual and three disks. The main program disk should be backed up on floppy or installed on the hard drive. Use any copy program to make a backup of the floppy or from WorkBench, simply drag the SkidMarks disk into an empty drawer on your HardDisk.

As configuration files are saved on the program disk, also ensure the disk is not write protected.

The track disks are copy protected and in no situation should copying these disks be attempted. Not only is this illegal but it will affect our sales, sales which will allow us to develop the game and others for your enjoyment. Large corporate money bags we are not, smalltime hardworking computer programmers we are. Please, do not pirate SkidMarks.

Either we continue working on more enhancements for the Amiga version including a track editor, car designer and super AGA version or we spend our time converting it to Japanese consoles, the choice is yours.

To run SkidMarks place the program disk in the Amiga's floppy drive and power up. Double click on the SkidMarks disk and then double click on the SkidMarks icon.

The main menu allows you to select the number of players, control methods and of course start the game. If you are playing one player mode with joystick control simply select START by pressing the joystick button or hit the RETURN key on the keyboard.

Options

Once SkidMarks has loaded you can choose to start a 1 or 2 player game, change your options or exit back to WorkBench. Use the joystick or the Up/Down/Return keys to select.

If you are playing SkidMarks between two computers select 1 or 2 players depending how many players are playing on your computer, the computer will then wait for the other computer to select 1 or 2 player mode. This way you can have 2,3 or 4 player linked play.

From the options screen you can choose from the following:

Select Car: after loading the select car screen you can then use fire to change the type of car and left/right to change colour. The number of different cars you can have will depend on the amount of memory in your computer and whether you are using the enhanced AGA cars. Select OK to return to the main Options screen.

Computer Team: there are four different computer teams to race against each with their own characteristics.

Difficulty: Novice Rooky and Pro will determine how fast SkidMarks will play and hence the reaction times required to master the tracks.

Game Mode: the three modes of play in SkidMarks are Practise, MatchRace and Championship. Practise lets the individual player race the tracks without competition (alone). MatchRace is for competing in one off races on any track while Championship lets the player participate in a 6 track series competing for points.

After the game has loaded the cars it will ask you to insert a track disk. SkidMarks comes with two track disks, more will be available shortly. From the track disk the options track options screen will be loaded.

If PRACTISE or MATCHRACE was selected from the main screen you will be able to select from any of the 6 tracks. When in CHAMPIONSHIP mode tracks must be raced in order.

Once you have selected a track the game will take a few seconds to load and then its race time.

Linking two Machines.

Using either a standard null modem cable between two Amigas (two male DB25 connectors with pins 2 and 3 crossed and pin 7 joined (up to 60 feet)) or via modem, SkidMarks can be played on two machines at once.

See the README file on the Program disk for details on connecting with the SkidSerial option.

Using SkidSerial up to 6 players can participate. While four drivers race (two on each machine) two others can act as managers via the chat-line raising stakes, developing strategy...

The true purpose of the chatline is however for sending abuse and profanities to opposing players, a macro key system is provided for off the cuff commentary.

General:

Hints & Tips

- Spend some time learning each track. Knowing the corners and finding the best approach to jumps is the best way to achieve good lap times.
- There are two strategies for good cornering. Either hug the the inside of the corner at all times or use the banking on the outside of the corner to increase speed.
- When taking jumps ensure that you are not turning. Also attempt to approach the jump so that when you land you are ready for the next corner.

If others are attempting to pass:

- Try and get them to ram you by blocking their way, such collisions will increase your own speed.
- Keep on the inside of the corner and attempt to push your opposition into the outside barrier.
- Accidentally spill your drink in their lap.
- If playing by modem hit them with a disconcerting message via the macro keys to destroy their concentration. It usually pays to make all your keyboard macros disconcerting to avoid mistaken pleasantries.

If you are losing badly:

- Concentrate on your own driving and learning the track rather than losing your cool.
- If playing locally use verbal obscenities to offput your opposition and perhaps spill more beer/fanta in their lap.
- If all else fails, stop, turn round and speed head on into the ain opposing traffic, always a good option if you are certain of being lapped.

Credits...

SkidMarks was developed in Blitz2 by Andrew Blackbourn.

Rodney Smith from Vision Software is responsible for all graphics in the game including the modelling of the 3 dimensional cars in Imagine and the detailing of the tracks in DPaint4.

Music was written by Anthony Milas.

Development and production were sponsored by Acid Software

Blitz2 was developed by Mark Sibly who is also responsible for the nasty two way road in one of the tracks.

Simon Armstrong is responsible for small amounts of machine code several tracks and together with Benoit Varasse bringing this game to market.

Techo bits:

The AGA enhancements feature superhires scrolling and hires 16 colour sprites as featured in the new Blitz2 Display library.

The 800 frames for car animation are compressed and stored in fastmem if possible. Decompression to sprite memory as well as shadow generation is done every alternate frame for each car.

Steering, acceleration and collision detection is also calculated in interleaved frames.

Communications are done at 50 cps (500 bps). This allows for 1 byte of communication per frame, two bits for ascii chat, 2 bits acceleration (single car) and 4 bits for steering (both cars). A two frame delay for remote machines is required for optimum performance.

Tracks are currently 1024x516 16 colour bitmaps. Cars are 32x32 4 colour sprites, AGA cars are 64x32 16 colour hires sprites. Frames were rendered as 32 rotations with 5 pitches and 5 rolls, each 800 frames took some 10 hours to render on an Amiga4000. Custom software and ADPro were used to create the final images.

Corners are calculated using Quadratic B-Splines. The linear algebra for calculating boundary collisions is listed at the end of this manual. This represents some 50% of the machine code in SkidMarks. Percentage of BASIC code in SkidMarks is some 95% in total.

Extract from SkidMarks source code for quadratic calculations:

```
Function .w TestNew{first.l,second.l}
Macro e SizeOf .quadratic\1(a0):End Macro
Macro pa SizeOf .parse\1(a1):End Macro
UNLK a4:MOVE.l d0,a0:MOVE.l d1,a1
MOVEQ #0,d6:MOVEQ #0,d7
BSR strcur2:BGT failsc0:MOVEQ #1,d6:failsc0:ADD.l #hquad,a0:NOT d7
BSR strcur2:BGT failsc1:ADDQ #2,d6:failsc1:ADD.l #hquad,a0
BSR strcur1:BGT failsc2:ADDQ #4,d6:failsc2:ADD.l #hquad,a0:NOT d7
BSR strcur1:BGT failsc3:ADDQ #8,d6:failsc3
MOVE d6,d0:RTS
;
strcur1
MOVE.l !pa{f},d0:SUB.l !e{lc},d0:ASR.l #5,d0 ;d0=ni
MOVE.l !pa{f},d1:SUB.l !e{lf},d1:ASR.l #5,d1 ;d1=nj
TST !e{straight}:BNE handlestraight1:BRA handlecurve
strcur2
MOVE.l !pa{f},d0:SUB.l !e{lc},d0:ASR.l #5,d0 ;d0=ni
MOVE.l !pa{f},d1:SUB.l !e{lf},d1:ASR.l #5,d1 ;d1=nj
TST !e{straight}:BNE handlestraight2
;
handlecurve: ;5 mults
MOVE.l !e{lb},d2:ASR.l #2,d2:MOVE.l !e{le},d3:ASR.l #2,d3 ;pb,pe
MULS d0,d2:MULS d1,d3:ADD.l d3,d2 ;d2=dot1
MOVE.l !e{la},d4:ASR.l #7,d4:MOVE.l !e{ld},d5:ASR.l #7,d5 ;pb,pe
MULS d0,d4:MULS d1,d5:ADD.l d5,d4 ;d4=dot2
SWAP d2:MULS d2,d2:ASL.l #2,d2 ;dot1*dot1
TST.w !e{lodd1}:BEQ obtuse:EXG d2,d4:obtuse:CMPL d4,d2:RTS
;
handlestraight1: ;2 mults
MOVE.l !e{le},d2:ASR.l #2,d2:MOVE.l !e{lb},d3:ASR.l #2,d3 ;pb,pe
MULS d0,d2:MULS d1,d3:TST d7:BNE noswap1:EXG d2,d3:noswap1:CMPL d2,d3:RTS
;
handlestraight2: ;2 mults
MOVE.l !e{le},d2:ASR.l #2,d2:MOVE.l !e{lb},d3:ASR.l #2,d3 ;pb,pe
MULS d0,d2:MULS d1,d3:TST d7:BEQ noswap2:EXG d2,d3:noswap2:CMPL d2,d3:RTS
;
End Function ;testnew
```

```
Function.w bouncenew {first.l,second.l}
UNLK a4:MOVE.l d0,a0:MOVE.l d1,a1
MOVE.l !pa{f},d0:SUB.l !e{lc},d0:ASR.l #5,d0 ;d0=ni
MOVE.l !pa{f},d1:SUB.l !e{lf},d1:ASR.l #5,d1 ;d1=nj
MOVE.l !pa{d},d2:ASR.l #2,d2 ;d2=b
MOVE.l !pa{dj},d3:ASR.l #2,d3 ;d3=d
MULS d2,d1:MULS d3,d0:SUB.l d0,d1 ;d1=top=ni.d-nj.b
MOVE.l !e{lb},d4:ASR.l #2,d4 ;d4=pb
MOVE.l !e{le},d5:ASR.l #2,d5 ;d5=pe
MULS d4,d3:MULS d5,d2:SUB.l d3,d2 ;d3=bot=pb.d-pe.b
SWAP d2:ASR.l #5,d1:TST d2:BEQ fail:succ?!?!?
DIVS d2,d1:EXT.l d1:ASL.l #8,d1 ;d1=t=any size
CMPL !e{lb1},d1:BLT fail:CMPL !e{lb2},d1:BGT fail
ASR.l #8,d1:MOVE d1,d0:MOVE d1,d2:MULS d4,d0:MULS d5,d2
ASR.l #6,d0:ASR.l #6,d2
ADD.l !e{lc},d0:ADD.l !e{lf},d2 ;d0=ti d2=tj
MOVE.l d0,d3:SUB.l !pa{f},d3:ASR.l #5,d3:MULS d3,d3
MOVE.l d2,d4:SUB.l !pa{f},d4:ASR.l #5,d4:MULS d4,d4:ADD.l d4,d3
```



```

MOVE.I !pa{di},d5      :ASR.I #5,d5:MULS d5,d5
MOVE.I !pa{dj},d6      :ASR.I #5,d6:MULS d6,d6:ADD.I d6,d5
CMP.I d5,d3:BPL fail
succ:MOVE.I d0,!pa{fi}:MOVE.I d2,!pa{fi}
    MOVE.w !e{langst},!pa{norm}
    MOVEQ#-1,d0:RTS
fail:MOVEQ#0,d0:RTS
End Function

```

```

Function.w bouncepod {zeroth.I,first.I,second.I}
UNLK a4:MOVE.I d0,a0:MOVE.I d1,a1:MOVE.I d2,a2
MOVE.I !pa{diri},d0:MOVE.I !pa{dirj},d1
TST.w !e{lodd3}:BEQ hoppy:NEG.I d0:NEG.I d1:hoppy
ASR.I #5,d0:MOVE d0,d4:ASR.I #5,d1:MOVE d1,d5 ;d1=d=d5 ;d0=b=d4
MOVE.I !pa{i},d2:SUB.I !e{lc},d2:ASR.I #6,d2 ;d2=a
MOVE.I !pa{j},d3:SUB.I !e{lf},d3:ASR.I #6,d3 ;d3=c
MOVE.I !e{ma},d6:ASR.I #1,d6:MULS d6,d1 ;1+5
MOVE.I !e{md},d7:ASR.I #1,d7:MULS d7,d0:SUB.I d0,d1 ;A=d1=ma.d-md.b
MOVE.I !e{lb},d6:ASR.I #2,d6:MULS d5,d6 ;2+5
MOVE.I !e{le},d7:ASR.I #2,d7:MULS d4,d7:SUB.I d7,d6 ;B=d6=lb.d-le.b
MULS d2,d5:MULS d3,d4:SUB.I d5,d4 ;5+6 d4=C bc-ad
MOVE.I d6,d0:SWAP d0:MULS d0,d0 ;B*B 14
MOVE.I d1,d2:SWAP d2:TST d2:BEQ fail2:SWAP d4:MULS d2,d4 ;17
ASL.I #5,d4:SUB.I d4,d0:TST.I d0:BMI fail2 ;d0=desc (14)
;d0=descr d1=A(6) d6=B(7) d4=C
MOVEQ#19-9,d2:MOVE.I #2048,d3:MOVEQ#0,d7
bloop:CMP.I d3,d0:BLT cewl
LSR.I #1,d0:ROXR#1,d7:LSR.I #1,d0:ROXR#1,d7
DBRA d2,bloop:cewl:ADD d0,d0
MOVEM 0(a2,d0),d0/d5:MULU d7,d5:NOT d7:MULU d7,d0
ADD.I d5,d0:LSR.I d2,d0
TST !e{lodd1}:BNE skipn:NEG.I d0:skipn:SUB.I d6,d0 ;(9)
MOVEQ#13-6,d7:ASL.I #2,d0 ;ANDREW MAKE = 15 (actually 9)
MOVEQ.I #0,d2:TST.I d1:BPL boop:NEG.I d1:NEG.I d0:boop
nxsh:ASR.I #1,d1:SUBQ#1,d7:MOVE d1,d2:CMP.I d1,d2:BNE nxsh
LSR #1,d1:TST d1:BEQ fail2 ;!?!?!?!
DIVS d1,d0:BVC noflo:MOVE#-1,$dff180:noflo
EXT.I d0:TST d7:BMI doasr:ASL.I d7,d0:BRA dunasr
doasr:NEG d7:ASR.I d7,d0:dunasr:MOVE.I d0,d3:ASL.I #6,d3
CMP.I !e{lb1},d3:BLT fail2:CMP.I !e{lb2},d3:BGT fail2
MOVE.I !e{ma},d1:MOVE.I !e{md},d2:ASR.I #2,d1:ASR.I #2,d2
MULS d0,d1:ASL.I #6,d1:SWAP d1:MULS d0,d2:ASL.I #6,d2:SWAP d2
MOVE.I !e{lb},d3:MOVE.I !e{le},d4:ASR.I #2,d3:ASR.I #2,d4
ADD.I d3,d1:MULS d0,d1:ASR.I #8,d1:ADD.I !e{lc},d1
ADD.I d4,d2:MULS d0,d2:ASR.I #8,d2:ADD.I !e{lf},d2
MOVE.I d1,d3:SUB.I !pa{i},d3:ASR.I #5,d3:MULS d3,d3
MOVE.I d2,d4:SUB.I !pa{j},d4:ASR.I #5,d4:MULS d4,d4:ADD.I d4,d3
MOVE.I !pa{di},d5 :ASR.I #5,d5:MULS d5,d5
MOVE.I !pa{dj},d6 :ASR.I #5,d6:MULS d6,d6:ADD.I d6,d5
CMP.I d5,d3:BPL fail2
succ2:MOVE.I d1,!pa{fi}:MOVE.I d2,!pa{fi}
    MOVE.I !e{langadd},d3
    MULS d0,d3:ASR.I #6,d3
    ADD.I !e{langst},d3:SWAP d3
    MOVE.w d3,!pa{norm}
    MOVEQ#-1,d0:RTS

fail2:MOVEQ#0,d0:RTS
End Function

```